International Journal of Computer and Communication Technology

Volume 1 | Issue 2 Article 4

April 2010

Text Categorization based on Associative Classification

Padmavati Shrivastava
M.Tech(CSE) IV Sem ,Reader,Dept. of CSE RIT , Raipur, India, padmavati.shrivastava@yahoo.co.in

Uzma Ansari
M.Tech(CSE) IV Sem ,Reader,Dept. of CSE RIT , Raipur, India, arshi.uzma@gmail.com

Recommended Citation

Shrivastava, Padmavati and Ansari, Uzma (2010) "Text Categorization based on Associative Classification," *International Journal of Computer and Communication Technology*: Vol. 1 : Iss. 2 , Article 4.

DOI: 10.47893/IJCCT.2010.1031
Available at: https://www.interscience.in/ijcct/vol1/iss2/4

Text Categorization based on Associative Classification

Padmavati Shrivastava[1], **Uzma Ansari**[2]
M.Tech(CSE) IV Sem,Reader,Dept. of CSE
RIT , Raipur, India
[1]padmavati.shrivastava@yahoo.co.in, [2]arshi.uzma@gmail.com

Abstract **-Text mining is an emerging technology that can be used to augment existing data in corporate databases by making unstructured text data available for analysis. The incredible increase in online documents, which has been mostly due to the expanding internet, has renewed the interest in automated document classification and data mining. The demand for text classification to aid the analysis and management of text is increasing. Text is cheap, but information, in the form of knowing what classes a text belongs to, is expensive. Text classification is the process of classifying documents into predefined categories based on their content. Automatic classification of text can provide this information at low cost, but the classifiers themselves must be built with expensive human effort, or trained from texts which have themselves been manually classified. Both classification and association rule mining are indispensable to practical applications. For association rule mining, the target of discovery is not pre-determined, while for classification rule mining there is one and only one predetermined target. Thus, great savings and conveniences to the user could result if the two mining techniques can somehow be integrated. In this paper, such an integrated framework, called associative classification is used for text categorization The algorithm presented here for text classification uses words as features , to derive feature set from preclassified text documents. The concept of Naïve Bayes classifier is then used on derived features for final classification.**

Categories and Subject Descriptors
H.2.8 [Database Applications]: Data mining

Index terms : Text Mining , Data Mining , Text Classification Association Rule Mining , Classification , Associative Classification , Naïve Bayes approach

I. INTRODUCTION

Data Mining is the discovery of hidden information found in databases and can be viewed as a step in the knowledge discovery process [1] [2]. One of the most important category of data mining applications is that of Text mining. **Text categorization** (a.k.a. text classification) is the task of assigning predefined categories to free-text documents. Text categorization or topic spotting) is the task of automatically sorting a set of documents into categories from a predefined set.It can provide conceptual views of document collections and has important applications in the real world. For example, news stories are typically organized by subject categories (*topics*) or geographical codes; academic papers are often classified by technical domains and sub-domains; patient reports in health-care organizations are often indexed from multiple aspects, using taxonomies of disease categories, types of surgical procedures, insurance reimbursement codes and so on. Another widespread application of text categorization is spam filtering, where email messages are classified into the two categories of *spam* and *non-spam*, respectively.

II TEXT CATEGORIZATION PROBLEM

Text Categorization TC may be formalized as the task of approximating the unknown *target function* $ø : D \times C \rightarrow \{T,F\}$ (that describes how documents ought to be classified, according to a supposedly authoritative expert) by means of a function $\acute{Ø}: D \times C \rightarrow \{T,F\}$ called the *classifier*, where $C = \{c_1, \dots, c_k\}$ is a predefined set of categories and D is a (possibly infinite) set of documents. If $ø(d_j, c_i) = T$, then d_j is called a *positive example* (or a *member*) of c_i, while if $ø(d_j, c_i) = F$ it

is called a *negative example* of c_i. The categories are just symbolic labels: no additional knowledge (of a procedural or declarative nature) of their meaning is usually available, and it is often the case that no metadata (such as e.g. publication date, document type, publication source) is available either. In these cases, classification must be accomplished only on the basis of knowledge extracted from the documents themselves.

However, when in a given application either external knowledge or metadata is available, heuristic techniques of any nature may be adopted in order to leverage on these data, either in combination or in isolation from the IR and ML techniques.

TC is a ***subjective*** task: when two experts (human or artificial) decide whether or not to classify document d_j under category c_i, they may disagree, and this in fact happens with relatively high frequency. A news article on **George W. Bush**

selling his shares in the Texas Bulls baseball team could be filed under **Politics**, or under **Finance**, or under **Sport**, or under any combination of the three, or even under neither, depending on the subjective judgment of the expert. Because of this, the meaning of a category is subjective.

A formal statement of the association rule problem is [3] [4]:

Definition 1: Text Classification

Let C = { c1, c2, ... cm} be a set of categories (classes) and D = { d1, d2, ... dn} a set of documents. The task of the text classification consists in assigning to each pair (ci, dj) of C x D (with 1 ≤ i ≤ m and 1 ≤ j ≤ n) a value of 0 or 1, i.e. the value 0, if the document dj doesn't belong to ci This mapping is sometimes refered to as the decision matrix:

	d_1	**...**	**d_j**	**...**	**d_n**
c_1	a_{11}	...	a_{1j}	...	a_{1n}
...	...	...	...	...	...
c_i	a_{i1}	...	a_{ij}	...	a_{in}
...	...	...	...	...	...
c_m	a_{m1}	...	a_{mj}	...	a_{mn}

III MOTIVATION FOR PRESENT WORK

The demand for text classification to aid the analysis and management of text is increasing. Text is cheap, but information, in the form of knowing what classes a text belongs to, is expensive. Text classification is the process of classifying documents into predefined categories based on their content. Automatic classification of text can provide this information at low cost, but the classifiers themselves must be built with expensive human effort, or trained from texts which have themselves been manually classified. Both classification and association rule mining are indispensable to practical applications. For association rule mining, the target of discovery is not pre-determined, while for classification rule mining there is one and only one predetermined target. Thus, great savings and conveniences to the user could result if the two mining techniques can somehow be integrated. In this paper, such an integrated framework, called associative classification is used for text categorization The algorithm presented here for text classification uses words as features , to derive feature set from preclassified text documents. The concept of Naïve Bayes classifier is then used on derived features for final classification.

IV ASSOCIATION RULE PROBLEM

A formal statement of the association rule problem is [3] [4]:

Definition 2: Let I ={I1, I2, … , Im} be a set of m distinct attributes, also called *literals*. Let D be a database, where each record (tuple) T has a unique identifier, and contains a set of items such that T⊆ I An *association rule* is an implication of the form X⇒Y, where X, Y⊆ I, are sets of items called *itemsets*, and X∩Y=Ø. Here, X is called antecedent, and Y consequent. Two important measures for association rules, support (s) and confidence (c), can be defined as follows.

Definition 3: The *support (s)* of an association rule is the ratio (in percent) of the records that contain X∪Y to the total number of records in the database. Support is the statistical significance of an association rule. Therefore, if we say that the support of a rule is 5% then it means that 5% of the total records contain X ∪Y.

Definition 4: For a given number of records, *confidence (c)* is the ratio (in percent) of the number of records that contain X∪ Y to the number of records that contain X. Thus, if we say that a rule has a confidence of 85%, it means that 85% of the records containing X also contain Y. The confidence of a rule indicates the degree of correlation in the dataset between X and Y. Often a large confidence is required for association rules.

The problem of mining association rules can be decomposed into two subproblems [5] as follows:

1) Find all sets of items which occur with a frequency that is greater than or equal to the user-specified threshold support, s.

2) Generate the desired rules using the large itemsets, which have user-specified threshold confidence, c.

Mining of association rules from a database consists of finding all rules that meet the user-specified threshold support and confidence.

V APRIORI ALGORITHM

The Apriori algorithm developed by [5] is a great achievement in the history of mining association rules [10]. It is by far the most well-known association rule algorithm. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as *candidate generation*, and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search to count candidate item sets efficiently. This algorithm uses prior knowledge of frequent itemsets properties, i.e. all nonempty subsets of frequent itemset must also be frequent. This property is called Apriori property.In Apriori terminology Ck - Candidate itemset of size k and Lk - Frequent itemset of size k.

Apriori algorithm has two steps –

[1] **Join step** : Ck is generated by joining Lk-1 with itself.

[2] **Prune step** : Any (k-1) itemset that is not frequent cannot be a subset of a frequent k itemset.

The Apriori generates the candidate itemsets by joining the large itemsets of the previous pass and deleting those subsets which are small in the previous pass without considering the transactions in the database. By only considering large itemsets of the previous pass, the number of candidate large itemsets is significantly reduced.In the first pass, the itemsets with only one item are counted. The discovered large itemsets of the first pass are used to generate the candidate sets of the second pass . Once the candidate itemsets are found, their supports are counted to discover the large itemsets of size two by scanning the database. In the third pass, the large itemsets of the second pass are considered as the candidate sets to discover large itemsets of this pass. This iterative process terminates when no new large itemsets are found.

Algorithmic features:
*Apriori is an efficient association rule mining algorithm which finds frequent itemsets using iterative level-wise approach based on candidate generation
*This algorithm uses prior knowledge of frequent itemsets properties, i.e. all nonempty subsets of frequent itemset must also be frequent.
*Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation, and groups of candidates are tested against the data.
*The algorithm terminates when no further successful extensions are found.
*Apriori uses breadth-first search and a hash tree structure to count candidate item sets efficiently.
*Uses a Level-wise search , where k-itemsets(An itemset that contains k items is a k- itemset)are used to explore (k+1)-itemsets, to mine frequent itemsets from transactional database for Boolean association rules.
Apriori [5] is one of the most popular data mining approaches for finding frequent itemsets from transactional datasets. The Apriori algorithm is the main basis of many other well-known algorithms and implementations. The main challenge faced by the researchers in frequent itemset mining has been to reduce the execution time. One of the best implementation of Apriori algorithm is published in [6]. We use Apriori algorithm as a means to extract the frequently occurring words in text documents used for training the classifier. We have not used Association rules between words that is word relation is not considered in this application.

VI NAÏVE BAYES THEOREM USED FOR CLASSIFICATION

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting.

The naive Bayes probabilistic model: Abstractly, the probability model for a classifier is a conditional model

$$p(C|F_1,\dots,F_n)$$

over a dependent class variable C with a small number of outcomes or classes, conditional on several feature variables F1 through Fn. Using Bayes' theorem, we can write

$$p(C|F_1,\dots,F_n)=\frac{p(C)\ p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}.$$

In plain English the above equation can be written as

$$\text{posterior}=\frac{\text{prior}\times\text{likelihood}}{\text{evidence}}.$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features Fi are given, so that the denominator is effectively constant. The numerator caan be rewritten as follows, using repeated applications of the definition of conditional probability:

$$p(C,F_1,\dots\dots F_n)\ = p(\ C\ )\ p(F_1|\ C)\ p(F_2|\ C\ ,\ F_1)\ p(F_3|\ C\ ,\ F_1\ ,F_2\ ,F_3)\dots\dots\ p(F_n|\ C\ ,\ F_1\ ,F_2\ ,F_{3,}\dots\dots\ F_{n-1})$$

Now the "naive" conditional independence assumptions come into play: assume that each feature Fi is conditionally independent of every other feature Fj for j ≠ i. This means that

$$p(F_i|\ C\ ,\ F_j) = \quad p(F_i|\ C\ )$$

and so the joint model can be expressed as

$$p(C,F_1,\dots\dots F_n) = p(C)\ p(F_1|\ C)\ p(F_2|\ C)p(F_3|C)\dots.$$

$$= p(C)\prod_{i=1}^{n} p(F_i|C).$$

Constructing a classifier from the probability model :

In the above discussion the independent feature model, that is, the naive Bayes probability model has been derived. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule. The corresponding classifier is the function classify defined as follows:

$$\mathrm{classify}(f_1,\dots,f_n)=\underset{c}{\mathrm{argmax}}\ p(C=c)\prod_{i=1}^{n}p(F_i=f_i|C=c).$$

VII METHOLOGY USED

Text Categorization Flowchart:

The Association based classification technique is used to classify documents from Reuters 21578 dataset collection. The main phases are

a. Training Dataset Preprocessing

b. Building the Associative Classifier

c. Testing the Associative Classifier

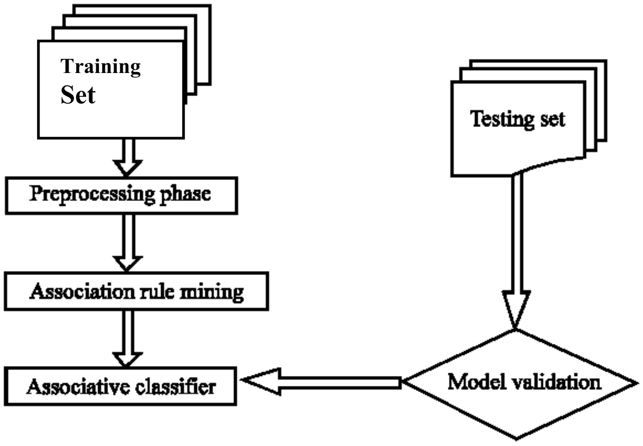

Fig. 1 Text Categorization Flowchart

Preprocessing: It involves the process of transforming the training dataset into a representation which is suitable for Apriori Algorithm.

The dataset used is the **Reuters-21578 dataset** with the standard "modApté" train/test split applied on it. These documents appeared on the Reuters newswire in 1987 and were manually classified by personnel from Reuters Ltd. I have used only three classes presently. The distribution of documents per class (used in this paper presently) is the following for **R52**:

R52			
Class	**# train docs**	**# test docs**	**Total # docs**
Orange	13	9	22
Rubber	19	1	20
Retail	31	9	40

Steps of Preprocessing:

First, all stop words in addition to periods, commas, and punctuations from the text are removed. Second, we delete all words other than frequent words. We define a word as frequent if it occurs more than once in a text. For counting a word whether it is frequent or not, some treatment for singular and plural form of a word is to be done. For this stemmed dataset is used(Porter Stemmer has been applied).Finally, the remaining frequent words are considered as a single transaction data in the set of database transaction. This process is applied to all text data before applying association mining to the transaction database.

Building the Associative Classifier:

The next step is to derive associated word set from training data. In this paper, total 63 numbers of news artciles are used as training data for learning to classify text from all three categories, of which 13 are from Orange class, 19 are form Rubber class and the rest 31 are from Retail news articles. After preprocessing the text data association rule mining is applied to the set of transaction data where each frequent word set from each abstract is considered as a single transaction.

Testing the Associative Classifier :

Large word sets generated in the training stage are used to classify a new text document. The text document to be classified is preprocessed (similar to the preprocessing steps adopted in the training stage)before comparing with the generated frequent wordsets using Apriori algorithm. Recalling the Naïve Bayes classifier for probability calculation.

$$\upsilon_{NB} = \text{argmax}\, P(\upsilon_j) \prod P(a_i \mid \upsilon_j)$$

The calculation for first term is based on the fraction of each target class in the training data. i.e. Prior probability for class Rubber , class Retail and category Orange.Then the second term of the equation is calculated by the following equation after adopting m-estimate approach [11] in order to avoid zero probability value,

$$\frac{n_k + 1}{n + |\text{vocabulary}|}$$

where, n = Total no of word set position in all training examples whose target value is

υ_j

n_k = No. of times the word set found among all the training examples whose target value is

υ_j

$|$vocabulary$|$ = The total number of distinct word set found within all the training data

VIII EXPERIMENTS CONDUCTED

Different tests have been conducted for classification of a new text document from the Reuters Dataset. To Tarin the classifier Apriori Algorithm for Association Rule Mining to generate frequent wordsets from preclassified documents. The experiments were conducted. The program has been developed using Visual Studio Dot Net Framework. Microsoft Visual C# 2008 Express edition and Microsoft Visual Basic 2008 Express edition have been used to develop the code. Microsoft Sql Server 2005 is used as backend.

VIII PARTIAL RESULTS

TRAINING THE CLASSIFIER:The following steps are followed to train the classifier

Step 1: Preprocessing

1)The periods, commas, and punctuations from the text are removed.

2)For e.g. if stemmed text from Orange class is:
"usda reduc citru estim program agricultur depart nation agricultur statist servic nass **will** chang citru estim program **for** california and arizona start nass will discontinu california forecast **for** lemon decemb februari march **and** june **and for** grapefruit **and** tangerin **for** month novemb forecast **for** lemon **will** issu octob novemb januari april **and** juli **and for** grapefruit octob januari april **and** juli **and for** tangerin octob januari **and** april **will** chang estim program **for** california orang arizona forecast **will** drop **for** lemon orang grapefruit **and** tangerin novemb decemb februari march **and** june forecast retain octob januari april **and** juli **will** estim program **for** citru texa florida reuter "
Then after removing stop words (The highlighted words in above text) the result is:
"usda reduc citru estim program agricultur depart nation agricultur statist servic nass chang citru estim program california arizona start nass discontinu california forecast lemon decemb februari march june grapefruit tangerin month novemb forecast lemon issu octob novemb januari april juli grapefruit octob januari april juli tangerin octob januari april chang estim program california orang arizona forecast drop lemon orang grapefruit tangerin novemb decemb februari march june forecast retain octob januari april juli estim program citru texa florida reuter"

3)From the resulting text we extract individual words retaining only those which occur more than once.

4)Now we apply the Apriori algorithm to extract large wordsets for user specified minimum threshold values.
For eg the following table gives the partial resulta of Apriori algorithm where support is .04 and confidence is 0.75.
The first column represents Class name followed by support and confidence value, the second column represents the large wordsets and the last column represents the support value.

Retail3675	februari,level,pct	6
Retail3675	ago,month,pct	6
Retail3675	earli,month,pct	6
Retail3675	februari,month,pct	7
Retail3675	increas,month,pct	6
Retail3675	januari,month,pct	7
Retail3675	februari,month,retail	6
Retail3675	increas,month,retail	6
Retail3675	januari,month,retail	7
Retail3675	februari,month,rose	6
Retail3675	januari,month,rose	6
Retail3675	ago,month,year	6
Retail3675	earli,month,year	6
Retail3675	februari,month,year	7
Retail3675	increas,month,year	6
Retail3675	januari,month,year	7
Retail3675	adjust,decemb,pct,retail	6
Retail3675	adjust,depart,pct,retail	6
Retail3675	adjust,pct,retail	6
Retail3675	ago,pct,retail	6
Retail3675	bill,depart,pct,retail	6
Orange4075	offici,orang	5
Orange4075	juic,orang	9
Orange4075	juic,offici	5
Orange4075	final,orang	5
Orange4075	final,juic	5
Orange4075	fcoj,orang	7
Orange4075	fcoj,juic	6
Orange4075	depart,orang	8
Orange4075	depart,juic	6
Orange4075	depart,final	5
Orange4075	depart,fcoj	5
Orange4075	citru,orang	6
Orange4075	citru,juic	5
Orange4075	Brazil,orang	7
Orange4075	Brazil,offici	5
Orange4075	Brazil,juic	7
Orange4075	Brazil,final	5
Orange4075	Brazil,fcoj	5
Orange4075	Brazil,depart	6
Orange4075	Brazil,depart,final	5
Orange4075	Brazil,depart,juic	6
Orange4075	Brazil,depart,orang	6
Orange4075	Brazil,fcoj,juic	5
Orange4075	Brazil,fcoj,orang	5
Orange4075	depart,fcoj,orang	5
Orange4075	Brazil,depart,final,juic	5
Orange4075	Brazil,final,juic	5
Orange4075	depart,final,juic	5
Orange4075	Brazil,depart,final,orang	5
Orange4075	Brazil,final,orang	5
Orange4075	depart,final,orang	5
Orange4075	Brazil,juic,offici	5
Orange4075	Brazil,depart,juic,orang	6
Orange4075	Brazil,fcoj,juic,orang	5
Orange4075	Brazil,final,juic,orang	5
Orange4075	Brazil,juic,orang	7
Rubber5075	agre,confer	10
Rubber5075	agre,agreement	10

Rubber5075	ad,week	10
Rubber5075	ad,rubber	15
Rubber5075	ad,reuter	15
Rubber5075	ad,produc	13
Rubber5075	ad,price	11
Rubber5075	ad,pact	10
Rubber5075	ad,nature	10
Rubber5075	ad,intern	10
Rubber5075	ad,dai	11
Rubber5075	agre,agreement,consum	10
Rubber5075	agre,agreement,inra	10
Rubber5075	agre,agreement,intern	10
Rubber5075	agre,agreement,nature	10
Rubber5075	agre,agreement,pact	10
Rubber5075	agre,agreement,produc	10
Rubber5075	agre,agreement,reuter	10
Rubber5075	agre,agreement,rubber	10
Rubber5075	agre,confer,pact	10
Rubber5075	agre,confer,rubber	10
Rubber5075	agre,consum,dai	10
Rubber5075	agre,agreement,consum,inra	10
Rubber5075	agre,consum,inra	11
Rubber5075	agreement,consum,inra	10
Rubber5075	agre,agreement,consum,intern	10
Rubber5075	agre,consum,intern	11
Rubber5075	agreement,consum,intern	10
Rubber5075	agre,agreement,consum,natur	10
Rubber5075	agre,consum,nature	11

Table 1: Word Sets generated using Apriori Algorithm

CLASSIFICATION PROCESS: To classify a new document the following steps are followed:

Suppose the document to be classified(after stemming) is : *“usda florida orang report consid bearish agricultur depart latest estim florida orang product and orang juic yield bearish for frozen concentr orang juic futur market yield increas greater expect fcoj trader and analyst usda project averag yield gallon fcoj per box versu last month estim gallon govern estim florida orang product exclud templ mln box versu mln last month templ unchang mln box trader and analyst unexpectedli larg yield increas outweigh anticip drop box count reuter”*

Step 1:
Since the data is already stemmed the stop words are only to be removed. This results in the following text:

“usda florida orang report consid bearish agricultur depart latest estim florida orang product orang juic yield bearish frozen concentr orang juic futur market yield increas greater expect fcoj trader analyst usda project averag yield gallon fcoj box versu month estim gallon govern estim florida orang product exclud templ mln box versu mln month templ unchang mln box trader analyst unexpectedli larg yield increas outweigh anticip drop box count reuter”

Step 2:
Next we extract individual words from the output of step 1 to collect those words which are frequent that is occurring more than once. The result is a set words wwith their corresponding occurrence frequency

analyst	2
bearish	2
box	4
estim	3
fcoj	2
florida	3
gallon	2
increas	2
juic	2
mln	3
month	2
orang	5
product	2
templ	2
trader	2
versu	2
yield	4

Table 2: The frequent wordsets of the document to be classified

Step 3:
The next step is to match the word sets formed from these words found in step 2 with the large word sets found using the Apriori algorithm:

increas,month	**increas,month,**pct,retail
juic,orang	brazil,depart,final,**juic,orang**
fcoj,orang	**fcoj,orang**
fcoj,juic	**fcoj,juic**
fcoj,juic,orang	**fcoj,juic,orang**

Step 4:
The next step is to find the probabilities for each class using Bayes Theorem:

Total no of wordsets : 2244 (more than one word)
45(Orange)+508(Retail)+1691(Rubber)=2244

Prior probability:

Orange =.0200
Retail = .2264
Rubber =.7536

The corresponding probabilities are:

Retail=.2264 * .0025*.00036*.00036*.00036*.00036=
0.00000000000000000950662656

Orange:.02*.00043*.0044*.0035*.0030*.0026=
0.000000000000001033032

Rubber=.7536*.00025*.00025*.00025*.00025.00025*=
0.0000000000000000007359375

Since the highest probability is that of Orange class this document is placed in Orange class

CONCLUSION

-Word Set of items two (at least) or more is generated from Association mining. So there is no option for considering a single word using association concept.

- Association mining largely reduces the number of words to be considered for classifying texts, keeping only words having association between them.
- Possibility of words common in more than one target classes is higher than the possibility of word set in more than one target classes. So considering a single word for classification increases the possibility of wrong classification.
- Considering word set instead of word for text classification increases the possibility of failure of text classification. But this possibility of failure can be reduced by considering increased number of training data.

ACKNOWLEDGMENT

Thanks to the faculties at RIT, Raipur for all the suggestions.
This work is being developed during the execution of the M.Tech Project(major).

REFERENCES

[1] Ming-Syan Chen, Jiawei Han and Philip S. Yu, Data Mining: An Overview from a Database Perspective, *IEEE Transactions on Knowledge and Data Engineering,* Vol. 8, No. 6, pp. 866-883, 1996.

[2] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth, From Data Mining to knowledge Discovery: An Overview, *Advances in Knowledge Discovery and Data Mining,* Edited by Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padraic Smyth, and Ramasamy Uthurusamy, AAAI Press, 1996, pp 1-34.

[3] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami, Mining Association Rules Between Sets of Items in Large Databases, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207-216, Washington, D.C., May 1993.

[4]David Wai-Lok Cheung, Vincent T. Ng, Ada Wai-Chee Fu, and Yongjian Fu,Efficient Mining of Association Rules in Distributed Databases, *IEEE Transactions on Knowledge and Data Engineering,* Vol. 8, No. 6, pp. 911-922, December 1996.

[5] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules, *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994, pp. 487-499.

[6] F. Bodon, "A Fast Apriori Implementation," In B. Goethals and M. J. Zaki, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Vol. 90 of CEUR Workshop Proceedings, 2003.

[10] David Wai-Lok Cheung, Vincent T. Ng, Ada Wai-Chee Fu, and Yongjian Fu, Efficient Mining of Association Rules in Distributed Databases, *IEEE Transactions on Knowledge and Data Engineering,* Vol. 8, No. 6, pp. 911-922, December 1996.

[11]Mitchell M. T., 1997. "Machine Learning", McGraw Hill, New York, 1997.

[12]David D. Lewis, 1992. "Feature Selection and Feature Extraction for Text Categorization, appeared in Speech and Natural Language", Proceedings of a workshop held at Harriman, New York, February 23-26, 1992. Morgan Kaufmann, San Mateo, CA.

[13]Eibe Frank, "Automatic Keyphrase Extraction", http://www.nzdl.org/kea/

Eibe Frank and Ian H. Witten, 2000. "Data Mining: Practical Machine Learning Larning Tools and Techniques with Java Implementation", Morgan Kaufmann Publisher: CA, 2000.

[14]Hayes, P. and Weinstein, S. 1990. "CONSTRUE/ TIS: a system for content-based indexing of a database of news stories", in IAAI-90, 1990.

[15]Jiawei Han and Micheline Kamber, 2001. "Data Mining: Concepts and Techniques", Morgan Kaufmann Publisher: CA, 2001.

[16]Lewis, D. and Croft, W., 1990. "Term clustering of syntactic phrases", in ACM SIGIR-90, PP. 385-404, 1990.

[17]www.cs.waikato.ac.nz/ml/weka